超音波非破壊評価に関係する逆問題の3次元時間域動弾性境界積分方程式法による数値解法について (微分方程式の数値解法と線形計算)

AUTHOR(S):

吉川, 仁; 西村, 直志

CITATION:

吉川, 仁 ...[et al]. 超音波非破壊評価に関係する逆問題の3次元時間域動弾性境界積分方程式法による数値解法について (微分方程式の数値解法と線形計算). 数理解析研究所講究録 2003, 1320: 89-100

ISSUE DATE:

2003-05

URL:

http://hdl.handle.net/2433/43082

数理解析研究所講究録 1320 巻 2003 年 89-100

超音波非破壊評価に関係する逆問題の 3 次元時間域動弾性境界積分方程式法による数値解法について

京都大学大学院・工学研究科　吉川仁 (Hitoshi Yoshikawa)
Graduate School of Eng., Kyoto Univ.
京都大学・学術情報メディアセンター　西村直志 (Naoshi Nishimura)
Academic Center for Computing and Media Studies, Kyoto Univ.

1 序論

著者らは、これまでレーザ計測により得られる実データからクラックの位置や形状を決定する問題を、時間域境界積分方程式法 (time domain boundary integral equation method)[1] を用いて解析する研究を行なってきている [2, 3, 4]。実験に用いているアルミニウム合金の S 波速度は約 3100m/sec であり、入射波の周波数は 500KHz 程度である。従って、扱う S 波の波長は約 6.2mm となる。そうすると、S 波の挙動を正確に表現するには 1 波長に多くの境界要素を含む必要がある事から、境界要素のサイズは一辺 1mm 程度となるので、数 cm 四方の領域を解析するのに数千〜数万自由度の分割が必要になる。また、時間ステップ幅も要素サイズに合わせて小さくとる必要があるため、数値解析は時間域の大規模問題となる。時間域の BIEM では、解が時間に関する畳み込み積分の形で表現される。ある時刻における解は、過去の全履歴の影響を受けるため、解析には多大な計算時間を要する。本報では、従来型の (多重極法によらない) 時間域 BIEM の計算効率を 2 つの面から改良する事を提案する。まず、畳み込み積分に関するアルゴリズムを改良し、無駄なメモリー使用や再計算を極力排除する方法を検討する。さらに、より多くのメモリーの確保と、計算時間の短縮のために、コードの並列化を検討する。

2 3 次元時間域動弾性問題における境界積分方程式法のアルゴリズム改良

2.1 積分方程式の定式化

時間域 BIEM のの例として、3 次元弾性体 D に存在する単一クラック S による入射波の散乱問題を考える。境界条件が全て表面力 t で与えられる場合、次の初期値境界値問題を未知の変位場 u について解く事になる。

$$\begin{aligned} &\mu\Delta\boldsymbol{u}+(\lambda+\mu)\nabla\nabla\cdot\boldsymbol{u}=\rho\ddot{\boldsymbol{u}} \ \text{ in } D\setminus S\times(t>0) \\ &\mathrm{T}\boldsymbol{u}=\boldsymbol{t} \text{ on } \partial D\times(t>0), \quad \boldsymbol{u}|_{t=0}=\dot{\boldsymbol{u}}|_{t=0}=0 \ \text{ in } D \\ &\mathrm{T}\boldsymbol{u}^{\pm}=0 \ \text{ on } S\times(t\geq 0), \quad \boldsymbol{\varphi}=\boldsymbol{u}^{+}-\boldsymbol{u}^{-}=0 \ \text{ on } \partial S \end{aligned} \tag{1}$$

ここに、λ,μ はラメ定数、ρ は密度、T はトラクション作用素、$+(-)$ はクラックの単位法線ベクトルの正（負）からの極限値、'$\dot{(\ )}$' は時間微分、$\boldsymbol{\varphi}$ はクラックの開口変位を表す。

式 (1) に対応する境界積分方程式は次のように書ける。

$$\begin{aligned} \frac{1}{2}\boldsymbol{u}(\boldsymbol{x},t) &= \int_{\partial D}\Gamma(\boldsymbol{x},\boldsymbol{y},t)*\mathrm{T}\boldsymbol{u}(\boldsymbol{y},t)dS-\text{v.p.}\int_{\partial D}\Gamma_I(\boldsymbol{x},\boldsymbol{y},t)*\boldsymbol{u}(\boldsymbol{y},t)dS \\ &+ \int_{S}\Gamma_I(\boldsymbol{x},\boldsymbol{y},t)*\boldsymbol{\varphi}(\boldsymbol{y},t)dS, \quad \boldsymbol{x}\in\partial D, \end{aligned} \tag{2}$$

$$
\begin{aligned}
0 &= \int_{\partial D} \mathrm{T}\Gamma(\boldsymbol{x},\boldsymbol{y},t) * \mathrm{T}\boldsymbol{u}(\boldsymbol{y},t)dS - \int_{\partial D} \mathrm{T}\Gamma_I(\boldsymbol{x},\boldsymbol{y},t) * \boldsymbol{u}(\boldsymbol{y},t)dS \\
&+ \text{p.f.} \int_S \mathrm{T}\Gamma_I(\boldsymbol{x},\boldsymbol{y},t) * \varphi(\boldsymbol{y},t)dS \quad \boldsymbol{x} \in S
\end{aligned} \tag{3}
$$

ここに、$\Gamma(\boldsymbol{x},\boldsymbol{y},t), \Gamma_I(\boldsymbol{x},\boldsymbol{y},t)$ はそれぞれ動弾性問題の基本解と二重層核、'$*$' は時間に関する畳み込み積分、v.p. は Cauchy の主値、p.f. は発散積分の有限部分を表す。

2.2 時間域 BIEM の従来の手法

時間域動弾性積分方程式を時間域の内挿関数 $M^\ell(t)$、空間域の内挿関数 $N^q(\boldsymbol{y})$ を用いて離散化する。Neumann 境界条件が与えられていれば、代数方程式は次の形で得られる。

$$
\begin{aligned}
&\sum_q A_{ij}^{pq}(\Delta t)\begin{bmatrix} u_j(\boldsymbol{x}^q, n\Delta t) \\ -\varphi_j(\boldsymbol{x}^q, n\Delta t)\end{bmatrix} + \begin{bmatrix} \frac{1}{2}u_i(\boldsymbol{x}^p, n\Delta t) \\ 0 \end{bmatrix} \\
&= b_i(\boldsymbol{x}^p, n\Delta t) - \sum_q \sum_{\ell=1}^{n-1} A_{ij}^{pq}((n+1-\ell)\Delta t)\begin{bmatrix} u_j(\boldsymbol{x}^q, \ell\Delta t) \\ -\varphi_j(\boldsymbol{x}^q, \ell\Delta t)\end{bmatrix}
\end{aligned} \tag{4}
$$

ここに、$\boldsymbol{x}^p \in \partial D$ の時、

$$
\begin{aligned}
A_{ij}^{pq}(\ell\Delta t) &= \int_{\partial D+S} \int \Gamma_{Iij}(\boldsymbol{x}^p, \boldsymbol{y}, \tau) M^\ell(\tau) N^q(\boldsymbol{y}) d\tau dS_y \\
b_i(\boldsymbol{x}^p, n\Delta t) &= \sum_q \sum_{\ell=1}^{n} (\mathrm{T}\boldsymbol{u})_j(\boldsymbol{x}^q, \ell\Delta t) \int_{\partial D} \int \Gamma_{ij}(\boldsymbol{x}^p, \boldsymbol{y}, \tau) M^{n+1-\ell}(\tau) N^q(\boldsymbol{y}) d\tau dS_y
\end{aligned}
$$

$\boldsymbol{x}^p \in S$ の時、

$$
\begin{aligned}
A_{ij}^{pq}(\ell\Delta t) &= \int_{\partial D+S} \int (\mathrm{T}\Gamma_I)_{ij}(\boldsymbol{x}^p, \boldsymbol{y}, \tau) M^\ell(\tau) N^q(\boldsymbol{y}) d\tau dS_y \\
b_i(\boldsymbol{x}^p, n\Delta t) &= \sum_q \sum_{\ell=1}^{n} (\mathrm{T}\boldsymbol{u})_j(\boldsymbol{x}^q, \ell\Delta t) \int_{\partial D} \int (\mathrm{T}\Gamma)_{ij}(\boldsymbol{x}^p, \boldsymbol{y}, \tau) M^{n+1-\ell}(\tau) N^q(\boldsymbol{y}) d\tau dS_y
\end{aligned}
$$

式 (4) から明らかなように、時刻 $n\Delta t$ の解 $u_i(\boldsymbol{x}^p, n\Delta t)$, $\phi_i(\boldsymbol{x}^p, n\Delta t)$ を得るには、時刻差が Δt から $n\Delta t$ までの影響係数 $A_{ij}^{pq}(\ell\Delta t)$ $(\ell = 1, \cdots n)$ が必要となる。一般に行なわれている BIEM による時間域問題の解析では、各時間ステップ $t = n\Delta t$ $(n = 1, \cdots, N_t)$ において、時間差が $n\Delta t$ の影響係数 $A_{ij}^{pq}(n\Delta t)$ の積分を計算し、メモリーへストアする。時間差が 1 から $n-1$ までの影響係数 $A_{ij}^{pq}(\ell\Delta t)$ $(\ell = 1, \cdots n-1)$ については、以前の時間ステップで既に計算しストアされているため、それらをメモリーから呼びだし行列・ベクトル積演算を行なう事で解くべき代数方程式を得る事ができる。 通常、3 次元時間域波動問題では、ソースからの波動が到達しない領域や波動の通り過ぎる領域では影響係数は 0 であり係数行列は疎となる事から、行列の非 0 成分のみを計算しストアすればよい。

時間ステップ N_t まで影響係数のストアが可能である時、影響係数の積分計算は各時間ステップで 1 回のみ行なえば良い。影響係数の積分計算は BIEM で最も計算時間を必要とする部分であるため、BIEM 解析に要する全計算時間は $O(N_t)$ となる。しかし、境界要素数や時間ステップ数の大きい大規模問題を取り扱う場合は、全ての影響係数をストアするだけのメモリーを確保できるとは限らない。時刻差が $t = N_m\Delta t$ $(N_m < N_t)$ までの影響係数しかストアできない場

合、$u_i(\boldsymbol{x}^p, n\Delta t), \varphi_i(\boldsymbol{x}^q, n\Delta t)$, $(n > N_m)$ を求める代数方程式を得るには、時刻 $t = n\Delta t$ $(n = N_m+1, \cdots, N_t)$ において影響係数 $A_{ij}^{pq}((N_m+1)\Delta t), \cdots, A_{ij}^{pq}(n\Delta t)$ を計算する必要がある。つまり、影響係数の積分計算を行なうルーチンは $t \leq N_m\Delta t$ では 1 回、$t \geq (N_m+1)\Delta t$ では $n - N_m$ 回、全体では $\dfrac{(N_t - N_m)(N_t - N_m + 1)}{2} + N_m$ 回実行される。大規模問題の場合、N_m は N_t に比べかなり小さくなる事が多い。そのため、結局全体の計算量はほぼ $O(N_t{}^2)$ となり、解析には多大な計算時間を要する事になる。以下、本報では時間域 BIEM の従来の手法を '素朴な時間域 BIEM' と呼ぶ。

本報では、従来型の (多重極法によらない) 時間域 BIEM の計算効率を 2 つの面から改良する事を提案する。まず、畳み込み積分に関するアルゴリズムを改良し、影響係数の再計算を極力排除する方法を検討する。次に、時間積分を離散化する際に用いられる内挿関数に関するアルゴリズムを改良し、影響係数の計算に要する時間の短縮を試みる。

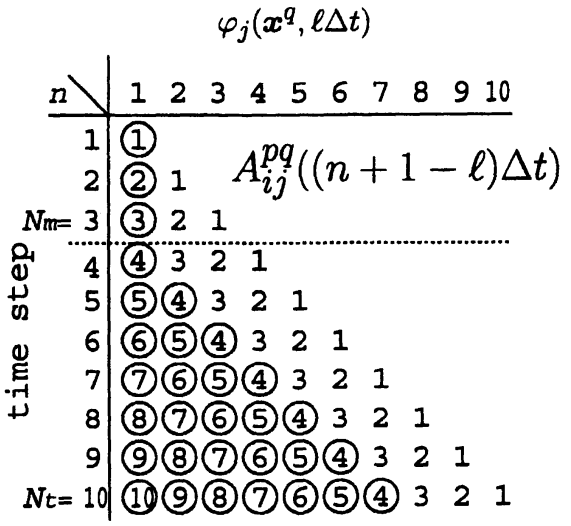

Fig. 1: 素朴な時間域 BIEM ($A_{ij}^{pq}(N_m\Delta t)$ までストア可能)

2.3 時間域 BIEM アルゴリズムの改良

2.3.1 畳み込み積分に関するアルゴリズムの改良

大規模 3 次元動弾性問題を実用的な時間で解くために、畳み込み積分に関するアルゴリズムを改良し、影響係数を計算するルーチンを実行する回数を減らす事を試みる。

時刻 $t = n\Delta t$ において計算される影響係数 $A_{ij}^{pq}(n\Delta t)$ は、時刻 $\ell\Delta t, (\ell = n, \cdots, N_t)$ において $u_j(\boldsymbol{x}^q, (\ell+1-n)\Delta t)$、$\varphi_j(\boldsymbol{x}^q, (\ell+1-n)\Delta t)$ (以下、繁雑さを避けるため、代表して φ_j のみ標記する) との行列・ベクトル積演算に用いられ、得られるベクトルは代数方程式の右辺へ加えられる。しかし、時刻 $t = n\Delta t$ において、1 つ前の時間ステップまでの解 $\varphi_j(\boldsymbol{x}^q, \ell\Delta t)$, $(\ell = 1, \cdots, n-1)$ は既に求められているため、$t = n\Delta t$ 以降の代数方程式の構成に必要となる行列・ベクトル積演算 $A_{ij}^{pq}(n\Delta t)\varphi_j(\boldsymbol{x}^q, \ell\Delta t)$, $(\ell = 1, \cdots, \min(n-1, N_t+1-n))$ を $t = n\Delta t$ の時点で行ない、それ

ぞれ時刻 $(\ell+n-1)\Delta t$ の代数方程式の右辺に加えておく事 (cast forward) が可能である。この考え方を用いれば、影響係数 $A_{ij}^{pq}(n\Delta t)$ $(n=[\frac{N_t+3}{2}]\cdots,N_t)$ (ここに $[a]$ はガウス記号であり、a を越えない最大の整数を表す) の計算は各時間ステップ $t=n\Delta t$ において 1 度行なうだけで良く、またストアする必要もない事がわかる。つまり、全計算ステップ数 N_t の問題において、時間差 $\Delta t,\cdots,N_t\Delta t$ の係数行列を全てストアする必要はなく、その半分の時間差 $\Delta t,\cdots,[\frac{N_t+1}{2}]\Delta t$ の係数行列のみストアするだけで良い。しかし、大規模問題の解析では、$N_m\Delta t<[\frac{N_t+1}{2}]\Delta t$ となる事が多い (ここに、$N_m\Delta t$ はストア可能な影響係数の最大の時間差である)。この場合、

1. 影響係数 $A_{ij}^{pq}(n\Delta t)$ $(n=1,\cdots,N_m)$ は時刻 $t=n\Delta t$ において計算しストアする。

2. $A_{ij}^{pq}(n\Delta t)$ $(n=N_m+1,\cdots,[\frac{N_t+1}{2}])$ については、$\varphi_j(\boldsymbol{x}^q,\ell\Delta t)$ $(1\le\ell\le N_t+1-n)$ との行列・ベクトル積演算が必要となる。しかし、時刻 $t=n\Delta t$ において計算される $A_{ij}^{pq}(n\Delta t)$ は $\varphi_j(\boldsymbol{x}^q,\ell\Delta t)$, $(1\le\ell\le n-1)$ との行列・ベクトル積演算しか実行できない。そのため、$A_{ij}^{pq}(n\Delta t)$ を時刻 $t=\{k(n-1)+1\}\Delta t$ (k は自然数で $k(n-1)+1\le N_t$) において計算し、$\varphi_j(\boldsymbol{x}^q,\ell\Delta t)$, $((k-1)(n-1)+1\le\ell\le\min(k(n-1),N_t+1-n))$ との行列・ベクトル積演算を行ない、それを時刻 $(\ell+n-1)\Delta t$ の代数方程式の右辺に加える。つまり、$A_{ij}^{pq}(n\Delta t)$ $(n=N_m+1,\cdots,[\frac{N_t+1}{2}])$ を計算するルーチンは全時間で $[\frac{N_t-1}{n-1}]$ 回実行される。

3. $A_{ij}^{pq}(n\Delta t)$ $(n=[\frac{N_t+3}{2}],\cdots,N_t)$ については、時刻 $t=n\Delta t$ において計算し、既知の解 $\varphi_j(\boldsymbol{x}^q,\ell\Delta t)$ $(1\le\ell\le\min(n-1,N_t+1-n))$ との行列・ベクトル積演算を行ない、得られたベクトルを時刻 $(\ell+n-1)\Delta t$ の代数方程式の右辺に加える。

この時、全時間で実行される影響係数の積分計算を行なうルーチンの回数は、

$N_m+\left[\frac{N_t}{2}\right]+\sum_{\ell=N_m+1}^{[\frac{N_t+1}{2}]}\left[\frac{N_t-1}{\ell-1}\right]$ 回となり、従来の大型問題の解析に比べてかなり減らす事ができる。

なお、畳み込み積分に関する行列ベクトル積演算を cast forward する改良を加えた時間域 BIEM のアルゴリズムにおいて、各時間ステップで行なわれる作業は、次の通りである。

1. $t=n\Delta t$ $(n=1,\cdots,N_m)$ の時、

   i) $A_{ij}^{pq}(n\Delta t)$ を計算し、メモリーにストアする。

   ii) メモリーから $A_{ij}^{pq}(\ell\Delta t)$ $(1\le\ell\le n-1)$ を呼びだし $\varphi_j(\boldsymbol{x}^q,(n+1-\ell)\Delta t)$ との行列・ベクトル積演算を行ない、$\varphi_j(\boldsymbol{x}^q,n\Delta t)$ を求める代数方程式を構成する。

2. $t=n\Delta t$ $(n=N_m+1,\cdots,N_t)$ の時、

   i) $k=\frac{n-1}{m-1}$ $(m=N_m+1,\cdots,n)$ が整数なら、$A_{ij}^{pq}(m\Delta t)$ を計算し、$\varphi_j(\boldsymbol{x}^q,\ell\Delta t)$, $((m-1)(k-1)+1\le\ell\le\min((m-1)k,\ N_t+1-n)$ との行列・ベクトル積演算を行ない、時刻 $(\ell+m-1)\Delta t$ の代数方程式の右辺に加える。

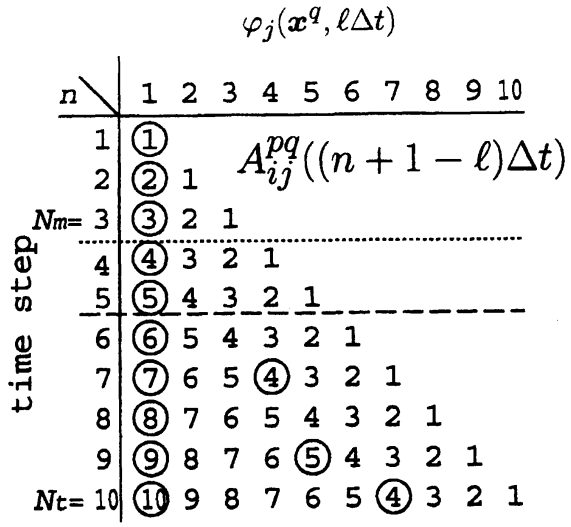

Fig. 2: 畳み込み積分に関して改良された時間域 BIEM

ii) メモリーから $A_{ij}^{pq}(\ell\Delta t)$ $(1 \le \ell \le N_m)$ を呼びだし $\varphi_j(x^q, (n+1-\ell)\Delta t)$ との行列・ベクトル積演算を行ない、$\varphi_j(x^q, n\Delta t)$ を求める代数方程式を構成する。

例えば、時間ステップ数が $N_t = 10$、時間差が $N_m\Delta t = 3\Delta t$ の影響係数までストアできるような問題では、素朴な方法 (Fig. 1) によれば影響係数の積分計算を行なうルーチンを 31 回呼ぶ必要があるが、提案する方法 (Fig. 2) を用いれば 13 回で済む。

2.3.2 時間変数の内挿関数に関するアルゴリズムの検討

次に、境界積分方程式の時間変数の離散化に用いられる内挿関数を構成するアルゴリズムについて検討する。なお以下では、本研究で用いる区分線形の時間内挿関数について述べる。

区分線形の時間内挿関数 $M^\ell(t)$

$$M^\ell(t) = \begin{cases} \dfrac{t-(\ell-2)\Delta t}{\Delta t} & (\ell-2)\Delta t \le t < (\ell-1)\Delta t \\ \dfrac{\ell\Delta t - t}{\Delta t} & (\ell-1)\Delta t \le t < \ell\Delta t \\ 0 & t < (\ell-2)\Delta t,\ t \ge \ell\Delta t \end{cases}$$

は、関数 $M_{\rm tri}{}^\ell(t) = \dfrac{\ell\Delta t - t}{\Delta t}$ により、次のように表される事に注意する。

$$M^\ell(t) = M_{\rm tri}{}^\ell(t) - 2M_{\rm tri}{}^{\ell-1}(t) + M_{\rm tri}{}^{\ell-2}(t) \tag{5}$$

式 (5) より、時間差 $n\Delta t$ の影響係数 $A_{ij}^{pq}(n\Delta t)$ は次の形で書ける。

$$A_{ij}^{pq}(n\Delta t) = A_{{\rm tri}\,ij}^{pq}(n\Delta t) - 2A_{{\rm tri}\,ij}^{pq}((n-1)\Delta t) + A_{{\rm tri}\,ij}^{pq}((n-2)\Delta t) \tag{6}$$

時間に関する積分を解析的に実行する場合、$A_{ij}^{pq}(n\Delta t)$ の計算は式 (6) を用いて行なう事になる。単純に考えれば、各時間ステップにおいて $A_{\mathrm{tri}}{}_{ij}^{pq}((n-1)\Delta t)$, $A_{\mathrm{tri}}{}_{ij}^{pq}((n-2)\Delta t)$, $A_{\mathrm{tri}}{}_{ij}^{pq}(n\Delta t)$ を計算し、式 (6) より $A_{ij}^{pq}(n\Delta t)$ を求めストアすれば良い。しかし、式 (6) より時刻 $n\Delta t$ において $A_{\mathrm{tri}}{}_{ij}^{pq}((n-1)\Delta t), A_{\mathrm{tri}}{}_{ij}^{pq}((n-2)\Delta t)$ の情報を持っていれば、$A_{\mathrm{tri}}{}_{ij}^{pq}(n\Delta t)$ の計算のみで影響係数を計算できる事は明らかである。そこで、各時間ステップにおいて $A_{\mathrm{tri}}{}_{ij}^{pq}(n\Delta t)$ の計算のみを行ない $A_{ij}^{pq}(n\Delta t)$ の代わりに、$A_{\mathrm{tri}}{}_{ij}^{pq}(n\Delta t)$ をストアする事にする。これにより各時間ステップにおける積分計算に要する時間は、単純に $A_{ij}^{pq}(n\Delta t)$ を求めてストアする手法に比べ約 $\frac{1}{3}$ に短縮される。しかし、影響係数 $A_{ij}^{pq}(n\Delta t)$ と過去の解 $u_j(\boldsymbol{x}^q, \ell\Delta t), \phi_j(\boldsymbol{x}^q, \ell\Delta t)$ との行列ベクトル積演算は、

$$A_{ij}^{pq}(n\Delta t)\begin{bmatrix} u_j(\boldsymbol{x}^q, \ell\Delta t) \\ -\varphi_j(\boldsymbol{x}^q, n\Delta t)\end{bmatrix} = A_{\mathrm{tri}}{}_{ij}^{pq}(n\Delta t)\begin{bmatrix} u_j(\boldsymbol{x}^q, \ell\Delta t) \\ -\varphi_j(\boldsymbol{x}^q, n\Delta t)\end{bmatrix} - 2A_{\mathrm{tri}}{}_{ij}^{pq}((n-1)\Delta t)\begin{bmatrix} u_j(\boldsymbol{x}^q, \ell\Delta t) \\ -\varphi_j(\boldsymbol{x}^q, n\Delta t)\end{bmatrix}$$
$$+A_{\mathrm{tri}}{}_{ij}^{pq}((n-2)\Delta t)\begin{bmatrix} u_j(\boldsymbol{x}^q, \ell\Delta t) \\ -\varphi_j(\boldsymbol{x}^q, n\Delta t)\end{bmatrix} \tag{7}$$

と分解される事ため、$A_{ij}^{pq}(n\Delta t)$ をストアする手法なら 1 回の行列ベクトル積演算を行なえば済むところを、$A_{\mathrm{tri}}{}_{ij}^{pq}(n\Delta t)$ をストアする手法では 3 回行なう必要がある。

2.4 数値例

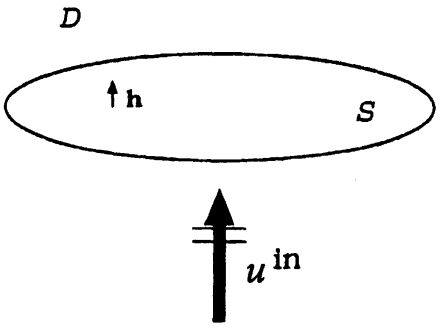

Fig. 3: 無限領域内のクラック散乱問題

3 次元時間域動弾性 BIEM の数値例として、無限弾性体 D に存在する半径 a の円形クラック S による弾性波動散乱問題を、従来の時間域 BIEM と提案した改良版時間域 BIEM を用いて解き、解析に要した CPU 時間を比較する。

入射波はクラックの法線方向（x_3 方向に一致）に進行する平面 P 波で $\sigma_{33} = p$ は一定とする。Poisson 比 $\nu = 0.25$、$\Delta t = 0.09a/c_T$ とし、11 ステップまでのクラックの開口変位を求める数値解析を行なった。数値解析を行なう上で、クラックを 2680 の境界要素 (未知数 8040) に分割している。得られた開口変位の第 3 成分の時間変動を Fig. 4 に示す。なお、使用した計算機は Alpha EV67 (666Mhz) を CPU とする Compaq 互換機であり、使用可能メモリーは 512MB である。

Fig. 1 に示した素朴な時間域 BIEM による解析では 11 ステップまでの開口変位を求めるのに 30 分 26 秒の CPU 時間を要したが、畳み込み積分について改良された BIEM で単純に A_{ij}^{pq} を計算する手法を用いた解析では 14 分 17 秒で計算が終了した。また、畳み込み積分と時間内挿関数に

ついて改良されたBIEMで$A_{\mathrm{tri}_{ij}}^{pq}$のストアを行なう手法では解析に7分35秒要した。本章で提案した改良版時間域BIEMによる解析を行なう事で、従来法に比べ計算時間が短縮できている。なお、ストアした影響係数の最大の時間差$N_m\Delta t$は、$N_m = 6$となり全時間ステップ$N_t = 11$の半分まで全てストア可能であった。

また、$N_t = 30$までのクラック開口変位を計算したところ、時間差$7\Delta t$までの影響係数のストアが可能であり($N_m = 7$)、素朴な時間域BIEMによる解析では446分4秒もの計算時間を必要としたが、畳み込み積分について改良されたBIEMで単純にA_{ij}^{pq}を計算する手法では58分30秒、畳み込み積分と時間内挿関数について改良されたBIEMで$A_{\mathrm{tri}_{ij}}^{pq}$のストア行なう手法では31分17秒で計算が終了した。この結果より、時間ステップの大きな問題の解析を行なう程、改良された時間域BIEMの有効性は顕著であると言えよう。

Table. 1: 計算時間(単一クラックによる弾性波動散乱問題)

解析手法	計算時間($N_t = 11$)	計算時間($N_t = 30$)
素朴なBIEM	30分26秒	446分4秒
改良版BIEM(A_{ij}^{pq} stored)	14分17秒	58分30秒
改良版BIEM($A_{\mathrm{tri}_{ij}}^{pq}$ stored)	7分46秒	31分17秒

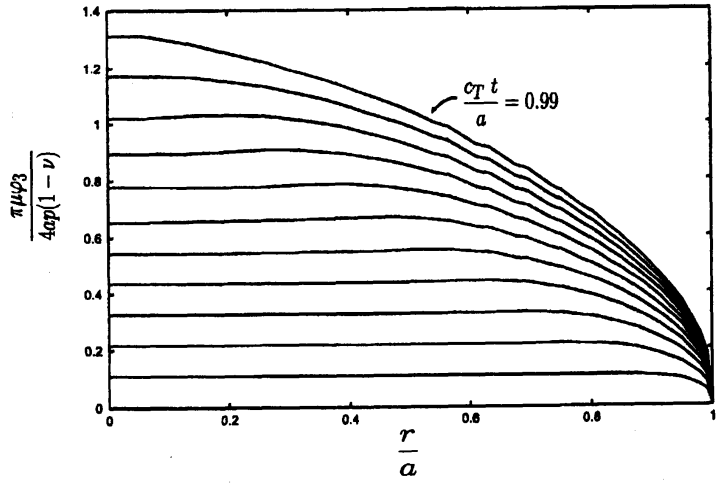

Fig. 4: クラックの開口変位

3 3次元時間域動弾性問題における境界積分方程式法の並列化

3.1 緒言

時間域のBIEMでは使用できるメモリーを増やし、より多くの影響係数をストアすれば、解析に要する計算時間は短縮される。そこで、MPI(Message-Passing Interface) [5]を用いて時間域BIEM

のコードを並列化し、複数の計算機のメモリーを使用する解析を考える。通常、BIEM解析では、離散化した積分方程式の影響係数の積分計算に多くの時間を要するため、BIEMに並列化を施す場合、影響係数の積分計算部分を並列化するのが最も効率が良いと考えられる。このため、時間域の境界積分方程式を離散化した際に各時間ステップで必要となる影響係数の計算、ストア、影響係数と過去の解との行列ベクトル積演算を複数台のプロセッサーで分割する並列計算を行なう。解析に要する計算時間は、積分計算や行列ベクトル積演算を複数台のプロセッサーに計算を分割する事により短縮され、また使用可能なメモリーが増える事によりさらに短縮される。

3.2 時間域BIEMの並列化アルゴリズム

時間域BIEMでは、代数方程式の係数行列は、解を求めるどの時間ステップにおいても時間差がΔtの影響係数行列 $A^{pq}(\Delta t)$ である (動弾性BIEMの代数方程式 (4) を参照)。影響係数 $A^{pq}(\Delta t)$ は、source 要素 S_q からの波動が 時刻 Δt で到達する領域に観測点 x_p が含まれるときのみ非0となる。ここで、 Courant 条件 より 波動が 1 ステップに 1 要素程度進むように Δt が定められているため、$A^{pq}(\Delta t)$ の非0成分はsource付近の観測点にのみ現われる。つまり、代数方程式の係数行列は、ほぼ diagonal のバンド型となり、時間域の境界積分方程式を離散化した代数方程式の求解には多くの計算時間を要しない。このため、影響係数の計算、ストア、行列ベクトル積演算を各プロセスで行ない、代数方程式の求解は1台のプロセスで行なう並列時間域BIEMのアルゴリズムを考える。なお、説明を簡略化するために、代数方程式を解くプロセスを'MASTER'、それ以外のプロセスを'SLAVE'と呼ぶ。

代数方程式の係数行列は、時刻 $t = \Delta t$ において計算される $A^{pq}(\Delta t)$ であるため、時刻 $t = \Delta t$ においては、'SLAVE'で計算された影響係数を'MASTER'へ送り、'MASTER'で係数行列をストアする。それ以外の時刻 $t = n\Delta t$ $(n = 2, \cdots, N_t)$ においては、影響係数と過去の解との行列ベクトル積の結果を'MASTER'へ送信し、それらの和を取り代数方程式右辺のベクトルを求める。その際、長さが未知数程度のベクトルの通信が必要となるが、時間域BIEMでは元来未知数が多くないのでデータ通信量は少なくて済む。なお、代数方程式の解法には繰り返し解法であるGMRESを用いる。GMRESにより、求められた解は'SLAVE'へ送り、次の時間ステップ以降の行列ベクトル積演算に用いられる。

3.3 数値例

並列時間域BIEMの数値例として、2.4の単一クラックによる弾性波動散乱問題を複数のプロセッサーを用いて解く。1台から8台のプロセッサーで並列計算を行ない、$N_t = 11$ までの開口変位を求めた。使用したプロセッサー台数毎の、計算時間、並列化効率、ストア可能な最大の影響係数 N_m を Table. 2 に、使用したプロセッサー台数と計算時間の関係を Fig. 5 の'×'印で示した。比較のために、並列化効率が 1 となる場合に相当する計算時間も Fig. 5 に'+'印で示した。なお、使用した計算機は全て CPU が Alpha EV67 (666Mhz) の Compaq 互換機で、使用可能メモリーは 512GB である。ネットワークは Myrinet を使用し、並列化のソフトウェアは mpich 1.2..4 を用いた。

Table. 2, Fig. 5 より、並列化により全体の計算時間が短縮され、並列化効率も 8 プロセッサーで 0.85 と高い値を示している事がわかる。これは、時間域のBIEMにおいて、計算時間を多く費やす影響係数の積分計算や、影響係数と過去の解との行列ベクトル積演算に並列化が施されて

いるためである。なお、時間ステップ $N_t = 11$ までの解析では、単一のプロセッサーの持つメモリーで、N_t の半分の時間差まで影響係数のストアが可能であるため、並列化により使用可能なメモリーが増える事による効果は現れていない。そこで、時間ステップ数を増やし $N_t = 30$ まで解析を行ない、使用したプロセッサー台数毎の計算時間、並列化効率、ストア可能な最大の影響係数 N_m を Table. 3 に、使用したプロセッサー台数と計算時間の関係を Fig. 6 の '×' 印で、並列化効率が 1 となる場合に相当する計算時間を Fig. 6 の '+' 印で示した。Table. 3, Fig. 6 に示した通り、並列化により使用可能なメモリーが増え、N_m が大きくなる事で影響係数の再計算が減り、全体の計算時間はさらに短縮されている。なお、本節で行なった解析は全て畳み込み積分と内挿関数についての改良を施した時間域 BIEM を用いて行なった。ここで、30 ステップまでのクラック開口変位を求める解析を例にとってみると、1 プロセッサーで '素朴な時間域 BIEM' による解析に要した計算時間 26764 秒 (Table. 1) が、畳み込み積分と内挿関数についての改良版時間域 BIEM と並列計算 (8 cpu) により 188 秒 (Table. 3) まで短縮されている。本報で提案する改良版並列時間域 BIEM が、非常に有効な手法である事を示す結果となっている。

Table. 2: wall time ($N_t = 11$)

プロセッサー数	1	2	3	4	5	6	7	8
wall time(sec)	466	241	163	124	100	87	73	67
並列化効率	1	0.94	0.93	0.92	0.91	0.87	0.89	0.85
N_m	6	6	6	6	6	6	6	6

Table. 3: wall time ($N_t = 30$)

プロセッサー数	1	2	3	4	5	6	7	8
wall time(sec)	1877	844	492	351	285	253	215	188
並列化効率	1	1.11	1.27	1.34	1.32	1.24	1.25	1.25
N_m	7	10	13	15	15	15	15	15

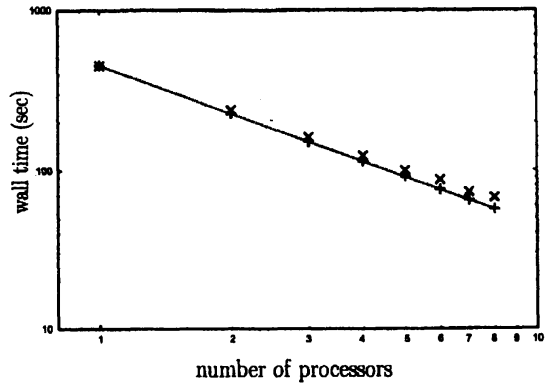

Fig. 5 プロセス数と計算時間 ($N_t = 11$)

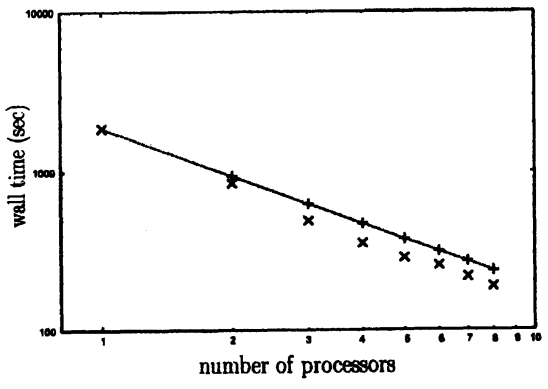

Fig. 6 プロセス数と計算時間 ($N_t = 30$)

4 大規模動弾性問題への適用例

畳み込み積分と内挿関数についての改良を施した並列時間域動弾性 BIEM を用いて、アルミ合金製供試体の表面クラック (Fig. 7) による弾性波散乱問題を解く。

弾性波の発生に用いるトランスデューサの供試体に及ぼす作用 (Fig. 8) をトラクションの境界条件とする Neumann 問題を解き、弾性波動に伴うスリット近傍 (Fig. 7 の R_2, R_3) の速度応答を計算した。ここで、P 波速度 $c_L = 6180$(m/sec)、S 波速度 $c_T = 3180$(m/sec)、$\Delta t = 0.075$ (μ sec) とした。また、境界要素数 10766(DOF: 32298)、時間ステップ数 130 の大規模問題を取り扱うため、解析には 8 台のプロセッサーを使用した。トランスデューサの中心から 26mm、及び 30mm 離れた点 (R_2, R_3) の応答に関して、得られた数値解と計測値を Fig. 9 に示す。これらは概ね良く一致していると言える。なお、解析に要した計算時間は 12 時間 50 分 1 秒であった。

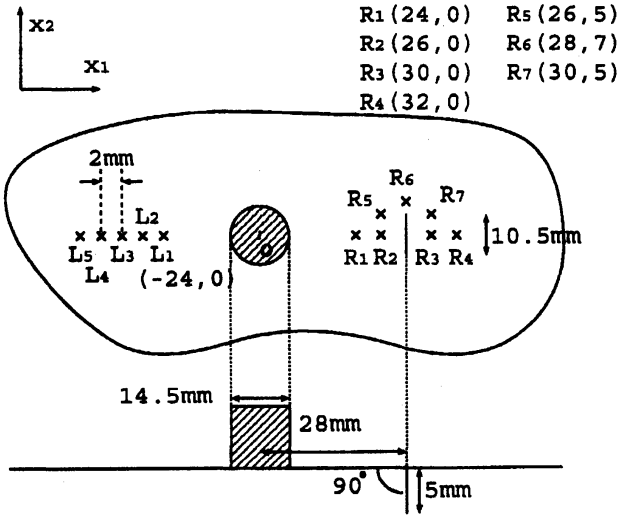

Fig. 7 計測設定

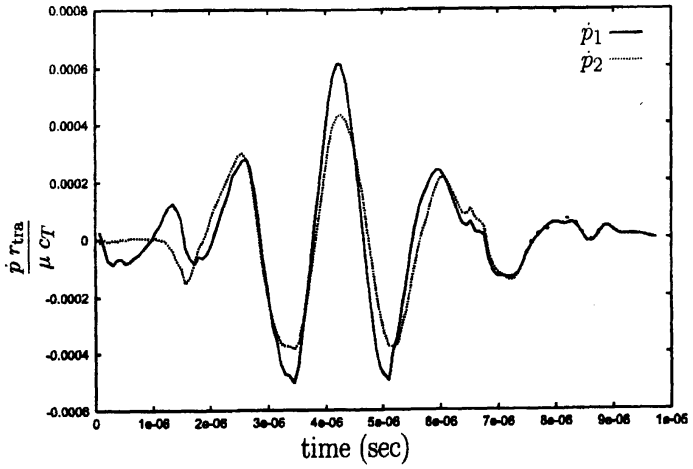

Fig. 8 $\dot{p}_1(t), \dot{p}_2(t)$

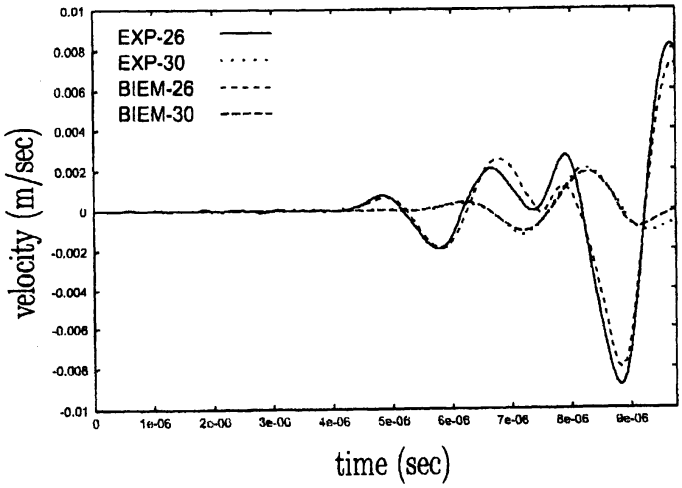

Fig. 9 スリット近傍の速度応答
(BIEM: 数値解, EXP: 計測値)

5 結論

時間域 BIEM の畳み込み積分と時間内挿関数に関するアルゴリズムの改良と MPI を用いた並列化により、各時間ステップにおける影響係数を全て記憶できないような大規模問題を、実用的な時間で解く事が可能となった。数値例として、単一クラックによる弾性波散乱問題を解き、提案する BIEM の有効性を示した。また、超音波非破壊評価に関連する問題に、改良版並列時間域動弾性 BIEM を適用したところ、良好な結果を得る事ができた。

参考文献

[1] 小林昭一, 福井卓雄, 北原道弘, 西村直志, 廣瀬壮一, 波動解析と境界要素法, (2000), 京都大学学術出版会.

[2] T. Kanbayashi, H. Yoshikawa, N. Nishimura and S. Kobayashi, Verification of ultrasonic transducer characteristics determined in an inverse problem based on laser measurements, *Inverse Problems in Engineering Mechanics II*,(Eds. M. Tanaka and G. S. Dulikravich), (2000), Elsevier, pp.327–332.

[3] 吉川仁, 西村直志, 小林昭一, On the determination of ultrasonic waves emitted from transducers using laser measurements with applications to defect determination problems, 土木学会応用力学論文集, **4**, (2001), pp.145–152.

[4] 吉川仁, 西村直志, 小林昭一, 3 次元時間域動弾性問題における境界積分方程式法のアルゴリズム改良と並列化, 土木学会応用力学論文集, **5**, (2002), pp.199–206.

[5] 青山幸也, 並列プログラミング虎の巻 MPI 版, 日本 IBM 株式会社, (1999).